

# The Snowball board

Chris Simmonds

*Embedded Linux Conference Europe 2011*

Copyright © 2011, 2net Limited

# Overview

- Description of the hardware
- Installing the tool chain
- Compiling the kernel
- Configuring the U-Boot environment
- Booting up
- What to do next

# Snowball “outside the box”



# Features

ARM® Dual Cortex A9 @ 1GHz,  
4 / 8GB e-MMC,  
1GB LP-DDR2,  
1x Micro-SD,  
1x RTC Battery Backup,  
1x HDMI Full HD,  
1x Ethernet 10/100Mbps,  
1x CVBS Video OUT,  
1x Audio OUT,  
1x Audio IN,  
1x USB OTG HS (480Mbps),  
1x DC IN +5V,  
1x Li-Ion Battery Charger,  
1x Serial Port (RS232 / Over USB  
1x JTAG Connector,  
1x MiPi34 Debug Connector,  
1x IEEE 802.11 b/g/n Wireless LA  
1 x Bluetooth 4.0/LEB(Ex. Antenna)  
1 x GPS (Ex. Antenna),  
1x 3 Axis Accelerometer,  
1x 3 Axis Magnetometer,  
1x 3 Axis Gyrometer,  
1x Pressure sensor (300-1100 mb  
3x Expansion Connectors (FSMC HSI, Audio, MiPi CSI / Camera, LC MiPi DSI, UART, SPI, I2C, GPIO)

# Linaro toolchain

- ARM v7a instruction set (for ARM Cortex A series)
- apt-get install .....

# Trying it out

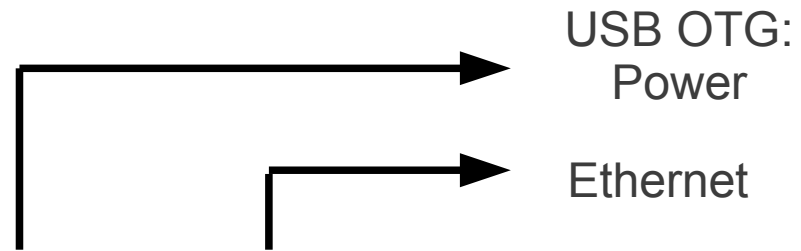
## The Hello World program

```
#include <stdio.h>
int main(void) {
    printf ("Hello, ARM\n");
    return 0;
}
```

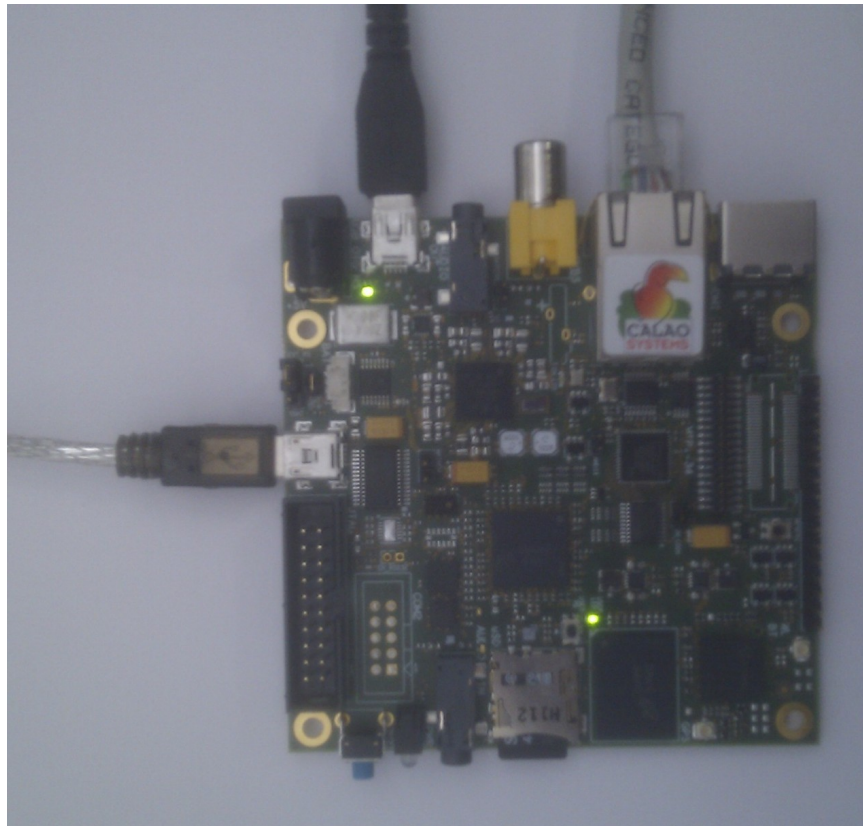
```
$ arm-linux-gnueabi-gcc hello.c -o hello-arm
```

```
$ file hello-arm
hello-arm: ELF 32-bit LSB executable, ARM, version 1 (SYSV),
dynamically linked (uses shared libs), for GNU/Linux 2.6.16, not stripped
```

# Connections



USB:  
serial console  
on ttyAMA2



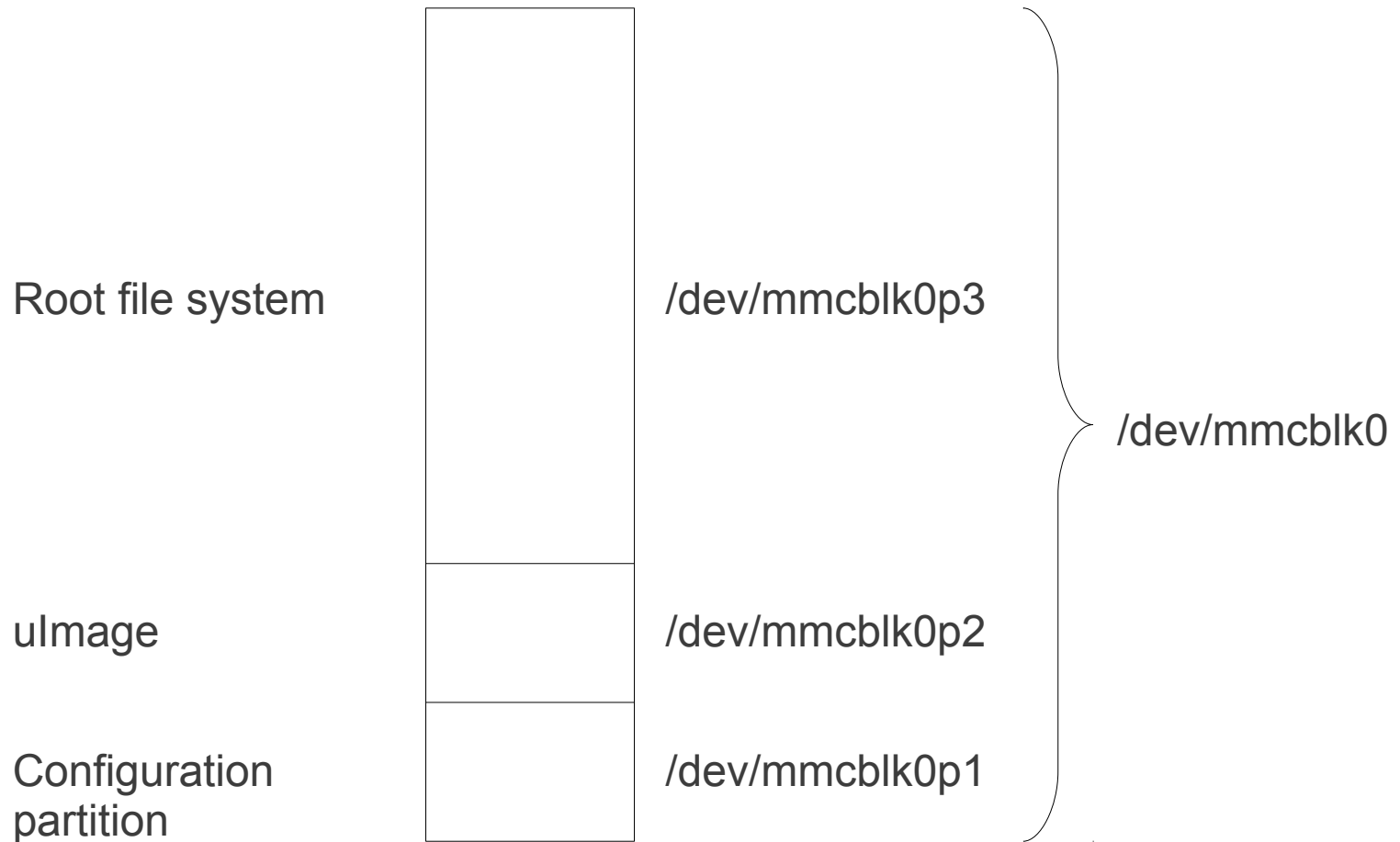
# Bootloader: U-Boot

- U-Boot version 2009.11
- Serial console over USB-serial adapter
  - 115200 8n1
- Minicom (terminal emulator) settings

```
+-----+
| A -   Serial Device       : /dev/ttyUSB0
| B - Lockfile Location    : /var/lock
| C -   Callin Program     :
| D -   Callout Program    :
| E -   Bps/Par/Bits       : 115200 8N1
| F - Hardware Flow Control : No
| G - Software Flow Control : No
|
|   Change which setting?
+-----+
```



# eMMC partitions



# Configuration partition

Contains a directory (TOC) and several software modules, see below. The rom bootloader scans for the ISSW module and then loads and executes it.

Image	Purpose
TOC	Table of contents. Tells the various boot steps where to find the next boot step
ISSW	Initial secure SW. This is the first code loaded by the boot rom.
X-LOADER	An intermediate boot loader which loads u-boot (or whatever is put in the NORMAL section).
MEM_INIT	Configures the memory controller. This module tends to be unique for each board revision.
PWR_MGT	Handles power management.
NORMAL	This is the final boot stage in the config partition. u-boot, by default.

# Hands-on

- Boot up from the USB sticks
- The instructions are in
  - `~/elce-2011/outside-the-box-lab-notes.pdf`
- Just follow the instructions...

# Further reading

- “Inner Penguin”
  - <http://www.embedded-linux.co.uk/>
- Embedded Linux Wiki
  - [http://elinux.org/Main\\_Page](http://elinux.org/Main_Page)
- Embedded Linux Primer
  - Christopher Hallinan, Prentice Hall
- Building Embedded Linux Systems, 2nd Edition
  - Karim Yaghmour et al, O'Reilly & Associates